Spotting Fake Reviews using Positive-Unlabeled Learning

Huayi Li[1], Bing Liu[1], Arjun Mukherjee[2], and Jidong Shao[3]

[1] Department of Computer Science,
University of Illinois at Chicago, Chicago, IL,
USA

[2] Department of Computer Science,
University of Houston, Houston,TX,
USA

[3] Dianping Inc., Shanghai,
China

hli47@uic.edu, liub@cs.uic.edu, arjun@cs.uh.edu, jidong.shao@dianping.com

Abstract. Fake review detection has been studied by researchers for several years. However, so far all reported studies are based on English reviews. This paper reports a study of detecting fake reviews in Chinese. Our review dataset is from the Chinese review hosting site Dianping[1], which has built a fake review detection system. They are confident that their algorithm has a very high precision, but they don't know the recall. This means that all fake reviews detected by the system are almost certainly fake but the remaining reviews may not be all genuine. This paper first reports a supervised learning study of two classes, fake and unknown. However, since the unknown set may contain many fake reviews, it is more appropriate to treat it as an unlabeled set. This calls for the model of learning from positive and unlabeled examples (or PU-learning). Experimental results show that PU learning not only outperforms supervised learning significantly, but also detects a large number of potentially fake reviews hidden in the unlabeled set that Dianping fails to detect.

Keywords. Fake reviews, Positive-Unlabeled learning, PU-learning.

1 Introduction

Opinions in reviews are increasingly used by individuals and organizations for making purchase decisions and for marketing and product design. Positive opinions often mean profits and fames for businesses and individuals, which, unfortunately, give strong incentives for imposters to post fake reviews to promote or to discredit some target products or services. Such individuals are called opinion spammers and their activities are called opinion spamming [10]. Detecting fake opinions is important to ensure that the online reviews continue to be trusted sources of opinions, rather than being full of fakes and lies.

In the past few years, several researchers have studied the problem. Existing studies are based on reviews in English. In this work, we perform a study on Chinese reviews. Our review dataset is from a popular review hosting site Dianping.com, which is the Chinese equivalent of Yelp.com. Dianping has built a system to detect fake reviews. It has been shown that the precision of the system is very high (due to the confidentiality agreement, we are unable to disclose the precise number), which means that when the system spots a fake review it is almost surely a fake review. We can trust the high precision due to two reasons. First, Dianping has a team of expert evaluators whose duty is to evaluate its detection algorithm. Every week, a random sample of detected fake reviews is manually evaluated by them based on all the data they collected (e.g., reviews, side information, IP addresses, click data, etc). Second, an even stronger evidence is

[1] http://www.dianping.com

ISSN 1405-5546
DOI: 10.13053/CyS-18-3-2035

that for each detected fake review, Dianping sends an email to its reviewer with reasons. Thus, we can trust the high precision of the system. However, Dianping does not know the true recall of their system because no one knows the exact number of fake reviews. High precision and unknown recall indicate that fake reviews detected by the system are almost certainly fake but the remaining reviews may not be all genuine, i.e., they may contain many fake reviews that Dianping's system cannot spot.

Dianping's algorithm is based on abnormal behaviors of reviews and their reviewers. No review text is used. In this paper, we focus on using review text content. The key advantage of using the text content is that it can detect fake reviews right after posting. Fake reviews thus will not cause any damage. A behavior based approach takes some time to accumulate evidences for detection.

Our data is a set of restaurant reviews from Dianping labeled with two classes, *fake* and *unknown*. A review in the unknown class means that the review has passed Dianping's algorithm, but it can still be fake. This paper performs two studies:

— **Supervised learning**: Using the labeled data, we first perform supervised learning to classify two types of reviews. Mukherjee et al. [21] performed this task using Yelp's filtered (fake) and unfiltered (non-fake) reviews [25, 18]. We perform it using Chinese reviews.

— **PU learning**: Since the unknown class can contain both fake reviews and non-fake reviews. The above classification is not entirely suitable. We thus treat the unknown class as unlabeled, which gives us a *positive and unlabeled* (PU) learning problem [3]. PU learning learns from positive (fake in our case) and unlabeled (or unknown) examples. Although [9] used a simple PU learning method to detect fake reviews, we show that methods proposed in our paper is significantly better.

Our experiments show that PU learning outperforms supervised learning significantly. What is even more important is that PU learning finds a large number of potentially fake reviews which have not been detected by Dianping's algorithm. This demonstrates the power of PU learning as its goal is to find hidden positives from the unlabeled set in the absence of negative training data.

2 Related Work

The main approach for opinion spam detection has been supervised learning. Although existing works have made important progresses, they mostly rely on ad-hoc fake and non-fake labels. In [10], duplicate and near duplicate reviews were assumed to be fake reviews in model building. The assumption, however, is too restricted for detecting general fake reviews. Li et al. [14] applied a co-training method on a manually labeled dataset of fake and non-fake reviews. This too may be unreliable as human labeling of fake reviews is quite poor [24]. Ott et al. [24] used Amazon Mechanical Turk (AMT) to get anonymous online workers to write 400 fake reviews on 20 popular Chicago hotels. 400 reviews from Tripadvisor.com on the same 20 Chicago hotels were used as non-fake reviews. They reported an accuracy of 89.6% using only word bigram features. [6] boosted the accuracy to 91.2% using some syntactic features. However, the AMT crafted fake reviews are not real fake reviews on real websites. The motivations and the psychological states of mind of the two types of reviewers/writers are very different, which can result in very different language styles [21].

Limited work has been done on detecting fake reviews using PU learning. Hernández et al. [9] proposed a simple PU learning framework called PU-LEA that iteratively removes positive training data from unlabeled data. However, they assume a continual but gradual reduction of the negative instances over iterations which unfortunately is not always true. We compare their model in the real-life datasets and found that our model outperforms it significantly.

Our work is most related to that in [21], which performed a supervised classification experiment on Yelp's filtered (fake) and unfiltered (non-fake) reviews. But in the above cases, PU learning was not used.

Apart from supervised learning, [11] studied unexpected review patterns, [26] and [1] investigated graph-based methods, Fei et al. [5] exploited review burstiness, Lim et al. [16] detected individual fake reviewers, [20] detected group fake reviewers, Xie et al. [28] did time-series analysis, Feng et al. [7] and Wu et al. [27] studied review rating distributions, and Li et al. [15] explored topic models.

Also related is the task of psycholinguistic deception detection which investigates lying words [8, 22], untrue views [19], computer-mediated deception in role-playing games [30], etc. However, fake reviews have very different dynamics. [24] showed that the features for detecting lies are not so effective for detecting fake reviews.

3 PU Learning Algorithms

As mentioned earlier, we will use supervised learning and PU learning to detect fake reviews. For supervised learning, we use Support Vector Machines (SVM) as it has been successfully applied to solve the problem in [21, 24]. As SVM is well known, we will not discuss it further. This section focuses on PU learning.

PU learning learns from a set of positive and unlabeled examples, where P denotes a set of positive examples, and U a set of unlabeled examples (which contains both hidden positive and hidden negative instances). The key characteristic is that it requires no negative training examples. The goal is to build a classifier using P and U to classify the data in U or a future test set T. In our setting, the test set T also acts as the unlabeled set U.

PU learning has been investigated by many researchers. A study of PAC learning under the statistical query model was given in [3]. [17] reported the sample complexity result to show how the problem may be solved. Subsequently, a number of practical algorithms [17, 29, 13] were proposed. They generally follow a two-step strategy: (1) identifying a set of reliable negative documents RN from the unlabeled set; and then (2) building a classifier using P (positive set), RN (reliable negative set) and U-RN (unlabeled set) by applying an existing learning algorithm iteratively. There are also some other approaches based on unbalanced errors [13, 4].

We used a publically available PU learning system, LPU[2] in our experiments. LPU uses a 2-step approach. There are three options (Spy, Roc, and NB) for step 1 and two options (EM and SVM) for step 2. We experimented with all combinations and found that using Spy in step 1 and EM or SVM in step 2 gives the best results. Below, we introduce these two combinations.

```
1: $RN \leftarrow \varnothing$;
   // Reliable negative set
2: $SP \leftarrow Sample(P, s\%)$;
   // Spy set
3: Assign each example in $P \setminus SP$ the class label $+1$;
4: Assign each example in $U \cup SP$ the class label $-1$;
5: $C \leftarrow NB(P \setminus SP, U \cup SP)$;
   // Produce a NB classifier
6: Classify each $u \in U \cup SP$ using $C$;
7: Decide a probability threshold $t$ using $SP$ and $l$;
8: for each $u \in U$ do
9:     if its probability $Pr(+|u) < t$ then
10:        $RN \leftarrow RN \cup u$
11:    end if
12: end for
```

Fig. 1. Spy algorithm for extracting RN from U

3.1 The Spy Algorithm: Step 1

Step 1 uses a spy technique to identify some reliable negatives (RN) from the unlabeled set U, which works as follows (Figure 1): First, a small set of positive examples (denoted by SP) called "spies" is randomly sampled from P (line 2). The default sampling ratio is $s = 15\%$ (Liu et al. 2002). Then, a Nave Bayes (NB) classifier C is built using $P \setminus SP$ as the positive set and $U \cup SP$ as the negative set (lines 3-5). The NB classifier is applied to classify each $u \in U \cup SP$, i.e., to assign a probabilistic class label $Pr(+|u)$ (+ means positive) to u. The idea of the spy technique is as follows. Since the spy examples are from P and are put into U as negatives in building the NB classifier, they should behave similarly to the hidden positives in U. We

[2] http://www.cs.uic.edu/~liub/LPU/LPU-download.html

```
1:  Each document in $P$ is assigned the class label $+1$;
2:  Each document in $RN$ is assigned the class label $-1$;
3:  Learn an initial NB classifier $f$ from $P$ and $RN$;
4:  do
        // E-Step
5:      for each document $d_i$ in $U \setminus RN$ do
6:          Using the current classifier $f$ to compute $Pr(c_j|d_i)$;
7:      end for
        // M-Step
8:      Learn a new NB classifier $f$ from $P$, $RN$ and $U \setminus RN$ using $Pr(c_j)$ and $Pr(w_t|c_j)$;
9:  while the classifier parameters stablize
10: The last iteration of EM gives the final classifier $f$;
11: for each document $d_i$ in $U$ do
12:     if its probability $Pr(+|d_i) \geq 0.5$ then
13:         Output $d_i$ as a positive document;
14:     else
15:         Output $d_i$ as a negative document;
16:     end if
17: end for
```

Fig. 2. EM algorithm with the NB classifier

thus can use them to find the reliable negative set RN from U. Using the probabilistic labels of spies in SP and an input parameter l (noise level), a probability threshold t is determined. Due to space constraints, we are unable to explain l. Details can be found in (Liu et al. 2002). t is then used to find RN from U (lines 8-12).

3.2 EM or SVM: Step 2

We discuss EM first. Given the positive set P, the reliable negative set RN, and the remaining unlabeled set $U \setminus RN$, we run EM [2] using the Naïve Bayes (NB) [23] as the base learning algorithm. Thus, EM basically runs NB iteratively until it converges.

The EM algorithm is given in Figure 2. Lines 1-3 build an initial NB classifier f using P and RN. Lines 4-9 apply f and build a new NB iteratively until convergence. Finally, the converged classifier is used to classify the unlabeled set U (lines 10-17). When SVM is used in the second step, it works similarly. See (Yu et al., 2002) for more details.

4 Empirical Evaluation

This section evaluates the supervised learning and PU learning approaches using real-life restaurant reviews from Dianping.com.

Dianping review dataset: Dianping has a filtering algorithm to filter fake reviews on their site. The algorithm has evolved over the years. As explained in the introduction, Dianping's review filter has a high precision but unknown recall.

Our experiment dataset consists of filtered (fake) reviews and unfiltered (unknown) reviews from 500 restaurants in Shanghai, China. Following [21, 24], we created a balanced dataset of 3476 fake (positive) reviews and 3476 unknown (negative) reviews. Due to the confidentiality agreement, we are unable to give the real proportion of fake reviews.

Since there are no white spaces between Chinese characters, we performed Chinese word segmentation using an existing segmentation tool called Jieba[3].

4.1 Supervised Learning Results

We report the results of supervised learning of two classes, *fake* (positive) and *unknown* (negative).

Classification settings: SVM (SVMLight [12]) is our learner. We use the linear kernel and all default parameters, which are also used in [21, 24]. All our results are obtained through 5-fold cross validation (CV).

Features: We use the standard unigrams and bigrams. Bigrams include unigrams. Unigrams and bigrams are based on words after segmentation. We also tried Chinese character n-grams, but they were poorer. For feature weighting, we use TF-IDF, which performs better than TF.

Evaluation measures: We use the standard *precision* (P), *recall* (R) and *F* score (F) because the user is mainly interested in the fake (positive) class. All the precision, recall and F score results are computed based on the positive (fake) class.

[3] https://github.com/fxsjy/jieba

Table 1. 5-fold CV results

	SVM			PU-LEA			Spy+EM			Spy+SVM		
	P	R	F	P	R	F	P	R	F	P	R	F
Unigrams	**0.54**	0.51	0.52	**0.54**	0.53	0.54	0.44	**0.86**	0.58	0.49	0.77	**0.60**
Bigrams	0.54	0.52	0.52	**0.55**	0.54	0.55	0.44	**0.89**	0.59	0.53	0.72	**0.61**

Classification results: The first column group of Table 1 gives the results of SVM for different feature sets. We can see that this is a very difficult problem. Unigrams and bigrams performed similarly. The difference is in the third digit after the decimal point. Compared with the F score of 0.72 using bi-grams on Yelp restaurant reviews in [21], the result here is much poorer. One reason is that Dianping reviews are much shorter than Yelp reviews and thus have less information for learners. On average, each Dianping review has 85.87 Chinese characters, and 59.63 words after segmentation, while each Yelp review has 130.60 words according to Yelp's data challege[4]. Another reason is that Chineses words are not naturally separated by white spaces. Errors produced by word segmentation would lead to poorer linguistic features.

4.2 PU Learning Results

In the PU learning experiments, we treat reviews in the unknown class as unlabeled reviews in training. In testing, we still treat them as unknown or negative. The second column group in Table 1 are results of PU-LEA [9] and the last two column groups correspond to the results of LPU(Spy+EM) and LPU(Spy+SVM). We can see that for both unigrams and bigrams our proposed PU learning is significantly better than SVM and PU-LEA in F scores based on paired *t-test* ($p < 0.001$). It is also important to note that the PU learning methods have much higher recalls but lower precisions. This may indicate that there are hidden fake reviews in the unknown set (see below).

Since the strength of PU learning is in uncovering hidden fakes in the unknown (unlabeled) set, we now explore that by first making some remarks about recall and precision in this context.

Recall: Since we do not know which reviews in the unknown set are fake, we cannot compute the precise recall. The recalls in the table are still based on the known fake reviews in the test set. However, we assume that the known fake reviews in the test set are representative of all fake reviews, including those hidden fakes in the unknown part of the test set. This is reasonable because Dianping's current method is entirely based on reviewer behaviors on their web site, and no text content is used in their filtering. This means that their method is independent of the review text content. Our classification uses text content only. Thus we can be reasonably confident that the recalls of the PU learning algorithms in Table 1 are good estimates of their true recalls.

Precision: Unfortunately, we cannot say the same about the precisions. The precisions in the table are based only on the known fake reviews, but do not cover any hidden fake reviews in the unknown class. There are two possible cases:

— There is no hidden fake review in the unknown set. In this case, the precisions in the table are the true precisions. However, this case is very unlikely because it means that Dianping has discovered all fake reviews.

— There are hidden fake reviews in the unknown set. This is more likely and this case is complicated. (a) If the classifier identified some of the hidden fake reviews (they are treated as false positives (FP) in the results of the table), then the true precision of the classifier will be higher. (b) If the classifier did not identify any hidden fake reviews in the unknown set, then the precisions are again the same as the ones given in the table. Intuitively, case (a) is more

[4] http://www.yelp.com/dataset_challenge

Table 2. Confusion matrix: *positive* is the *fake* class and *negative* is the *unknown* class.

	Classified positive	Classified negative
Labeled positive	TP	FN
Labeled negative	FP	TN

likely. Below, we give some strong evidences to show that this is the case.

4.3 Behavioral Analysis of False Positives

Precision and recall are computed based on the confusion matrix in Table 2, which has four cells: true positive (TP), true negative (TN), false positive (FP), and false negative (FN). Since both PU learning methods have significantly higher recalls than SVM (Table 1), we want to see whether the low precision is caused by hidden fake reviews in the unknown (negative in Table 2) class. Precision is defined as in Equation 1:

$$Precision = TP/(TP + FP) \tag{1}$$

We want to see whether some reviews in the FP set are potentially fake (i.e., true positives). Since it has been shown that by reading the reviews, it is very hard to spot fake reviews [24], we resort to abnormal behaviors of reviewers. We use two behavior clues to provide evidences:

— **Average number of reviews per day (ANR)**: ANR of a reviewer is computed by dividing the total number of reviews from him/her by the number of his/her active days. By active days, we mean in each of those days the reviewer has written at least one review. This is a good clue because if one writes a large number of reviews per day, he/she is suspicious.

— **Maximum content similarity (MCS)**: In order to save time and effort, a fake reviewer may reuse a past review or make some minor changes. We compute MCS for each reviewer using cosine similarity for every pair of his/her reviews. The rationale is that if one copies one's own reviews or makes minor changes to them, the reviewer is suspicious. A genuine reviewer expressing true experiences is unlikely to copy an old review and post it for a new restaurant.

To compute these two clues, we need the reviews from the reviewers on other restaurants. Dianping then provided us with a much larger set of reviews: 199,902 reviews on 78,669 restaurants.

From Table 1, we can see that Bigram features give slightly better results, we now analyze the results of Bigrams. To decide which FP reviews may be moved to TP, we use the MCS threshold of 0.8 and then vary the threshold of ANR from 2 reviews per day to more than 6 per day. Table 3 shows the number of false positives moved to true positive. Column #FP1 gives the number of FP reviews whose reviewers meet the MCS threshold and column #FP2 gives the number of FP reviews whose reviewers meet the ANR threshold in each row. Column #MV gives the number of FP reviews moved to TP (fake) as they satisfy either MCS $\geq$ 0.8 or ANR $\geq$ 2, 3, 4, 5, or 6. The updated results (averaged over 5-fold CV) for SVM, LPU (Spy+EM) and LPU (Spy+SVM) are shown in Table 4. Precisions of each method are changed because of the label adjustment but recalls remain the same. We cannot be sure that the recall also increases because we do not know how many reviews in TN may be positive (fake) too. F scores are computed by the new precisions and old recalls. We can see that the precisions of LPU (Spy+EM) and LPU (Spy+EM) increase markedly and so are their F scores, compared to those in Table 1. Changes for SVM and PU-LEA are also smaller than our proposed methods because they fail to capture those hidden fake reviews. Iterations in PU-LEA terminate before enough positives are identified from the unlabeled set.

To validate our results, we discussed our results with Dianping engineers. They agreed that those moved reviews should be fake (true positive) which their classifier cannot catch. We would like to stress that our analysis here is just to give some evidences that some of the FP reviews may actually be TP cases. There might also be other fake reviews in the FP set that our clues cannot find.

Table 3. Label adjustments by moving false positive (FP) to true positive (TP). MCS ≥ 0.8 is used for all experiments

	SVM			**PU-LEA**			**LPU (Spy+EM)**			**LPU (Spy+SVM)**		
ANR	#FP1	#FP2	#MV	#FP1	#FP2	#MV	#FP1	#FP2	#MV	#FP1	#FP2	#MV
≥ 2	49	0	49	41	0	41	170	228	295	86	114	149
≥ 3	49	0	49	41	0	41	170	110	227	86	56	115
≥ 4	49	0	49	41	0	41	170	62	201	86	31	101
≥ 5	49	0	49	41	0	41	170	43	192	86	22	97
≥ 6	49	0	49	41	0	41	170	34	185	86	17	94

Table 4. Results using bigrams after moving false positive (FP) to true positive (TP). MCS ≥ 0.8 is used for all experiments

	SVM			**PU-LEA**			**Spy+EM**			**Spy+SVM**		
ANR	P	R	F	P	R	F	P	R	F	P	R	F
≥ 2	0.63	0.52	0.57	0.62	0.54	0.58	0.59	**0.89**	**0.71**	**0.68**	0.72	0.70
≥ 3	0.63	0.52	0.57	0.62	0.54	0.58	0.55	**0.89**	0.68	**0.64**	0.72	**0.68**
≥ 4	**0.63**	0.52	0.57	0.62	0.54	0.58	0.53	**0.89**	0.66	**0.63**	0.72	**0.67**
≥ 5	**0.63**	0.52	0.57	0.62	0.54	0.58	0.52	**0.89**	0.66	0.62	0.72	**0.67**
≥ 6	**0.63**	0.52	0.57	0.62	0.54	0.58	0.52	**0.89**	0.65	0.62	0.72	**0.67**

5 Conclusion

This paper studied Chinese fake review detection. It makes two main contributions. First, this is the first reported study of opinion spam detection of Chinese reviews. Second, it used PU learning for the task as the unknown set is really an unlabeled set rather than the non-fake reviews set.

We have to learn from a set of positive (fake) and unlabeled examples. We have shown that PU learning has some major advantages. It not only outperforms the classic supervised learning SVM, but also more importantly, detects a large number of potential fake reviews hidden in the unlabeled set, which demonstrates the power of PU learning for solving the problem.

Acknowledgments

The authors would like to thank the spam detection team in Dianping for sharing the Chinese review dataset. This research paper is made possible through the help and support from engineers and scientists in Dianping who provided valuable suggestions and indispensable efforts in evaluation.

References

1. **Akoglu, L., Chandy, R., & Faloutsos, C.** (**2013**). Opinion fraud detection in online reviews by network effects. In *ICWSM*.
2. **Dempster, A. P., Laird, N. M., & Rubin, D. B.** (**1977**). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society, series B*, 39(1), 1–38.
3. **Denis, F.** (**1998**). PAC learning from positive statistical queries. In *ALT*. 112–126.
4. **Elkan, C. & Noto, K.** (**2008**). Learning classifiers from only positive and unlabeled data. In *KDD*. 213–220.
5. **Fei, G., Mukherjee, A., Liu, B., Hsu, M., Castellanos, M., & Ghosh, R.** (**2013**). Exploiting burstiness in reviews for review spammer detection. In *ICWSM*.

6. **Feng, S., Banerjee, R., & Choi, Y.** (**2012**). Syntactic stylometry for deception detection. In *ACL (2)*. 171–175.

7. **Feng, S., Xing, L., Gogar, A., & Choi, Y.** (**2012**). Distributional footprints of deceptive product reviews. In *ICWSM*.

8. **Hancock, J. T., Curry, L. E., Goorha, S., & Woodworth, M.** (**2007**). On lying and being lied to: A linguistic analysis of deception in computer-mediated communication. *Discourse Processes*, 45(1), 1–23.

9. **Hernández, D., Guzmán, R., Móntes y Gomez, M., & Rosso, P.** (**2013**). Using PU-learning to detect deceptive opinion spam. In *Proc. of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. Association for Computational Linguistics, Atlanta, Georgia, 38–45.

10. **Jindal, N. & Liu, B.** (**2008**). Opinion spam and analysis. In *WSDM*. 219–230.

11. **Jindal, N., Liu, B., & Lim, E.-P.** (**2010**). Finding unusual review patterns using unexpected rules. In *CIKM*. 1549–1552.

12. **Joachims, T.** (**1999**). Making large scale SVM learning practical.

13. **Lee, W. S. & Liu, B.** (**2003**). Learning with positive and unlabeled examples using weighted logistic regression. In *ICML*, volume 3. 448–455.

14. **Li, F., Huang, M., Yang, Y., & Zhu, X.** (**2011**). Learning to identify review spam. In *Proc. of IJCAI International Joint Conference on Artificial Intelligence*, volume 22. 2488.

15. **Li, J., Ott, M., & Cardie, C.** (**2013**). Identifying manipulated offerings on review portals. In *EMNLP*. 1933–1942.

16. **Lim, E.-P., Nguyen, V.-A., Jindal, N., Liu, B., & Lauw, H. W.** (**2010**). Detecting product review spammers using rating behaviors. In *Proc. of the 19th ACM international conference on Information and knowledge management*. ACM, 939–948.

17. **Liu, B., Lee, W. S., Yu, P. S., & Li, X.** (**2002**). Partially supervised classification of text documents. In *ICML*, volume 2. Citeseer, 387–394.

18. **Lowe, L.** (**2010**). http://officialblog.yelp.com /2010/03/yelp-review-filter-explained.html.

19. **Mihalcea, R. & Strapparava, C.** (**2009**). The lie detector: Explorations in the automatic recognition of deceptive language. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*. Association for Computational Linguistics, 309–312.

20. **Mukherjee, A., Liu, B., & Glance, N.** (**2012**). Spotting fake reviewer groups in consumer reviews. In *Proc. of the 21st international conference on World Wide Web*. ACM, 191–200.

21. **Mukherjee, A., Venkataraman, V., Liu, B., & Glance, N. S.** (**2013**). What yelp fake review filter might be doing? In *ICWSM*.

22. **Newman, M. L., Pennebaker, J. W., Berry, D. S., & Richards, J. M.** (**2003**). Lying words: Predicting deception from linguistic styles. *Personality and social psychology bulletin*, 29(5), 665–675.

23. **Nigam, K., McCallum, A. K., Thrun, S., & Mitchell, T.** (**2000**). Text classification from labeled and unlabeled documents using em. *Machine learning*, 39(2-3), 103–134.

24. **Ott, M., Choi, Y., Cardie, C., & Hancock, J. T.** (**2011**). Finding deceptive opinion spam by any stretch of the imagination. In *ACL*. 309–319.

25. **Stoppelman, J.** (**2009**). http://officialblog.yelp.com /2009/10/why-yelp-has-a-review-filter.html.

26. **Wang, G., Xie, S., Liu, B., & Yu, P. S.** (**2011**). Review graph based online store review spammer detection. In *IEEE 11th International Conference on Data Mining (ICDM)*. IEEE, 1242–1247.

27. **Wu, G., Greene, D., Smyth, B., & Cunningham, P.** (**2010**). Distortion as a validation criterion in the identification of suspicious reviews. In *Proc. of the First Workshop on Social Media Analytics*. ACM, 10–13.

28. **Xie, S., Wang, G., Lin, S., & Yu, P. S.** (**2012**). Review spam detection via temporal pattern discovery. In *Proc. of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 823–831.

29. **Yu, H., Han, J., & Chang, K. C.-C.** (**2002**). PEBL: positive example based learning for web page classification using SVM. In *Proc. of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 239–248.

30. **Zhou, L., Shi, Y., & Zhang, D.** (**2008**). A statistical language modeling approach to online deception detection. *IEEE Transactions on Knowledge and Data Engineering*, 20(8), 1077–1081.

Huayi Li is a direct PhD student at the University of Illinois at Chicago with Professor Bing Liu. He received his B.S degree of Computer Science from Nanjing Normal University, China. Previously he

had internships in Analysis and Experimentation in Microsoft, Map Analysis Center of Excellence in Nokia HERE Map and Chinese Academy of Science where he did various big data projects ranging from sentiment analysis, text mining, machine learning and A/B testing. Now he is also a research specialist in Health Media Collaboratory in UIC. His Ph.D. thesis is about collective classification in heterogeneous networks and topic modeling.

Bing Liu is a professor of Computer Science at the University of Illinois at Chicago (UIC). He received his PhD in Artificial Intelligence from the University of Edinburgh. Before joining UIC, he was a faculty member at the National University of Singapore. His current research interests include sentiment analysis and opinion mining, data mining, machine learning, and natural language processing (NLP). He has published extensively in top conferences and journals. He is also the author of two books: Sentiment Analysis and Opinion Mining (Morgan and Claypool) and Web Data Mining: Exploring Hyperlinks, Contents and Usage Data (Springer). In addition to research impacts, his work has also made important social impacts. Some of his work has been widely reported in the press, including a front-page article in The New York Times. On professional services, Liu has served as program chairs of many leading data mining related conferences of ACM, IEEE, and SIAM: KDD, ICDM, CIKM, WSDM, SDM, and PAKDD, as associate editors of several leading data mining journals, e.g., TKDE, TWEB, DMKD, and as area/track chairs or senior technical committee members of numerous NLP, data mining, and Web technology conferences. He currently also serves as the Chair of ACM SIGKDD, and is an IEEE Fellow.

Arjun Mukherjee is currently an Assistant Professor at the University of Houston. He obtained his Ph.D. from the University of Illinois at Chicago. He has been an intern fellow at Microsoft Research and Indian Statistical Institute. He has been supported various scholarships such as Deans Scholar, Chancellors Fellow and Provost and Deiss Fellow. His research spans several areas such as Bayesian inference, statistical data mining and natural language processing, machine learning, and social and information sciences with a particular emphasis on solving big-data problems in social media and the Web. His works have addressed a wide variety of social media problems including (1) modeling trust, reputation, opinion spam, deception and user behaviors (e.g., collusion, social-interactions, burstiness, etc.); (2) fine-grained latent variable modeling of sentiments expressed in online communications such as debates, reviews, and comments; and (3) market prediction and financial modeling using social sentiments. His works have been published in leading publication venues in Computer Science like KDD, WWW, ACL, EMNLP, CIKM, IJCAI, AAAI-ICWSM.

Jidong Shao is currently the director of the Credibility Group at Dianping.com, which mainly focuses on anti-spam, anti-fraud and business rating. He received his Ph.D. from the Zhejiang University, China. Before joining Dianping, he was a senior data mining expert at Alibaba.com. His main research interests include data mining, machine learning and their applications in internet business.

Article received on 22/08/2014; accepted on 18/09/2014.